

REMARKS

Introduction

All pending claims 1-48 are canceled without prejudice. Claims 49-89 are new. For the reasons discussed in detail below, all of the new claims are in condition for allowance.

Prior Art Rejections

The Examiner has rejected claims 1-48 under 35 U.S.C. § 103(a) over U.S. Patent No. 5,469,573 to McGill, III et al ("McGill") in view of U.S. Patent 5,745,669 to Hugard et al ("Hugard"). Applicants respectfully traverse these rejections. In the following, Applicants provide an overview of their invention and then discuss the differences with McGill and Hugard.

Applicants' technique is generally directed towards automatic backup and restoration of a failed system by capturing and storing the state of executing programs in addition to backing up the actual operating system and data files. Applicants' technique provides an application programming interface that a backup program may invoke for capturing the state of executing programs, including application programs. The information of a program's execution state is persistently stored on a medium along with the state of a hardware device such as a hard disk, catalog information of unmodifiable files, and registry information. Upon system failure, a recovery process may retrieve the previous state of the hard disk persistently stored and restore the hard disk state. The recovery

process may then retrieve the information of a previous execution state of programs from persistent storage and restore the programs. Advantageously, the application programming interface may be used by any third party developer for backing up application programs for restoration upon failure. Note that the above description is for example and informational purposes only, and should not be used to interpret the claims, which are discussed below.

McGill, in general, is very different from the present invention. First of all, McGill is directed to a data backup and restoration technique for restoring a configured operating system without reloading and reconfiguring the operating system from its original distribution media. To do so, McGill describes backing up the configured operating system by copying the operating system loaded on a computer system to a first media or backup device and copying configuration-specific data files such as system configuration files and device drivers to a second media or bootable recovery diskette. The configured operating system may then be subsequently restored on the computer system by initializing the computer system from the bootable recovery diskette and using the system configuration files and device drivers stored on the recovery diskette to provide a temporary operating system for loading a recovery application program from the backup device that, in turn, loads the operating system, configuration files and device drivers from the backup device. After loading is complete, the computer system is reinitialized using the loaded operating system. Once the computer system is restarted, any other backed up files can be restored to the hard drive.

In contrast to the claims of the present invention, McGill has simply no concept of storing the state of executing programs. Nor does McGill have any concept of providing application programming interface for capturing the state of executing programs as generally recited in Applicants' claims. First, independent claim 49 recites the limitation of "invoking an application programming interface for collecting information of an execution state of a program." Applicants' technique may provide a backup application programming interface for collecting information on programs and processes in use on the system and whose state is changing. McGill does not disclose any such limitation, but is instead concerned with restoring a configured operating system without reloading and reconfiguring the operating system from its original distribution media. Consistent with this stated purpose, McGill simply describes backing up the configured operating system by copying the operating system loaded on a computer system to a backup device. Nowhere in McGill can there be found any discussion of providing an application programming interface for capturing software execution state.

Second, independent claim 66 of the present invention recites "representing information of the execution state of a program." Applicants' technique may collect information on programs and processes in use on the system and whose state is changing. Again, McGill does not disclose any such limitation. Rather, consistent with the stated purpose, McGill merely describes backing up the configured operating system by copying the operating system loaded on a computer system to a backup device so that it may be restored

without reloading and reconfiguring the operating system from its original distribution media. Third, independent claim 76 recites "restoring the execution state of at least one program using the recorded recovery information of the execution state of the program". Applicants' technique further uses the information of the execution state of programs and processes to restore a computer system after failure. McGill does not disclose the limitation of claim 76. Moreover, McGill does not describe any capability to record information of a previous execution state of a program, nor could McGill without previously collecting information on programs and processes in use on the system including their execution state. Nowhere in McGill can there be found any discussion of capturing software execution state or retrieving such a captured software execution state.

Hugard, in general, is also significantly different from the present invention. Hugard is directed to providing a back-up utility that is simple to use for generally monitoring changes in the system configuration files. A back-up copy of the system configuration files may be made on the hard disk for comparison with the existing configuration files whenever the computer system is started. The utility indicates to users when changes are detected in the configuration files and provides the option to restore the saved version of the configuration files. The utility also allows detecting missing or changes application files by comparing the name, size, modification date and attributes of the application files with a previous version of the application files.

Like McGill, Hugard has simply no concept of storing the state of executing programs. Nor does Hugard have any concept of providing application programming interface for capturing the state of executing programs as generally recited in Applicants' claims. Rather Hugard describes monitoring a list of applications programs by comparing file status information. Hugard detects files that have changed and allows the recovery utility to replace any modified application program files with the backup copy of those files. Nowhere in Hugard can Applicants find a description of storing the state of executing programs, nor can Applicants find any concept of providing an application programming interface for capturing the state of executing programs as generally recited in Applicants' claims.

For instance, independent claim 49 recites the limitation of "invoking an application programming interface for collecting information of an execution state of a program." Significantly, Applicants' technique may provide a backup application programming interface for collecting information on programs and processes in use on the system and whose state is changing. Hugard does not disclose any such limitation, but is instead describes monitoring a list of applications programs by comparing file status information with previously save versions of the file. In specific, the utility described in Hugard allows for detecting missing or changes application files by comparing the name, size, modification date and attributes of the application files with a previous version of the application files. Nowhere in Hugard can there be found any discussion of

providing an application programming interface for capturing software execution state.

As another example, independent claim 66 of the present invention recites "representing information of the execution state of a program." Applicants' technique may collect and store information on programs and processes in use on the system and whose state is changing. Again, Hugard does not disclose any such limitation. Rather, Hugard may detect a change to an application programs by comparing file status information with previously save versions of the file. Moreover, independent claim 76 recites "restoring the execution state of the at least one program using the recorded recovery information of the execution state of the program". Applicants' technique further uses the information of the execution state of a programs and processes to restore the environment of a computer system after failure. Hugard does not disclose the limitation of claim 76. Furthermore, Hugard does not describe any capability to record information of a previous execution state of a program, nor could Hugard restore the environment of the computer system, including the execution state of application programs, without previously collecting such information. Nowhere in Hugard can there be found any discussion of capturing software execution state or retrieving such a captured software execution state.

For at least these significant reasons, applicants submit that the claims are patentable over the prior art of record, including McGill and Hugard, either together or alone.

Further, by law, in order to modify a reference to reject claimed subject matter, there must be some teaching or suggestion outside of Applicants' teachings to do so. Neither McGill nor Hugard have any such teachings or suggestions as to any such modification, let alone any teaching or suggestion as to how either system could be modified, or why it might be desirable to do so. In specific, the motivation described by McGill for his technique is to restore a configured operating system without reloading and reconfiguring the operating system from its original distribution media. The only other use McGill contemplates for his technique is for efficient factory loading of a fully configured operating system and application software into similarly configured workstations on a production line (McGill, column 2, lines 6-17.) Thus, McGill offers no motivation for modifying his invention to reach Applicants' recited limitations of "providing an application programming interface for capturing software execution state," or for "receiving information of the execution state of at least one program executing on the system." Nor does Hugard offer any motivation for modifying his invention to reach Applicants' claims. The motivation described by Hugard for his technique is to detect changes made to a backed-up version of files and prompt the user for restoring the backed-up version of the files. Hugard only maintains complete back-up files for the system configuration files and does not maintain complete back-up copies of application files but only notifies the user of changed or missing application files which the user may then reinstall or repair.

Thus, the only way in which McGill or Hugard could be modified to reach Applicants' claimed invention is via Applicants' own teachings, which is impermissible by law.

For at least these additional reasons, Applicants submit that the claims are patentable over the prior art of record, including McGill and Hugard, either together or alone.

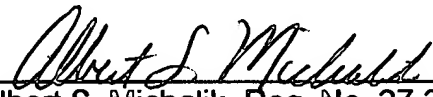
Regarding the dependent claims, Applicants respectfully submit that dependent claims 50-65, 67-75 and 77-89, by similar analysis, are not anticipated by McGill or Hugard, either together or alone. Each of the dependent claims 50-65 includes the limitation of "invoking an application programming interface for collecting information of an execution state of a program." Each of the dependent claims 67-75 includes the limitation of "representing information of the execution state of a program." And each of the dependent claims 77-89 includes the limitation of "restoring the execution state of the at least one program using the recorded recovery information of the execution state of the program." In addition to the limitations noted above, each of these dependent claims include additional patentable elements. For at least these reasons, Applicants submit that all of the dependent claims are in condition for allowance.

Conclusion

In view of the foregoing remarks, it is respectfully submitted that claims 49-89 of the present application are patentable over the prior art of record, and that the application is in good and proper form for allowance. A favorable action on the part of the Examiner is earnestly solicited.

If in the opinion of the Examiner a telephone conference would expedite the prosecution of the subject application, the Examiner is invited to call the undersigned attorney at (425) 836-3030.

Respectfully submitted,


Albert S. Michalik, Reg. No. 37,395
Attorney for Applicants
Law Offices of Albert S. Michalik, PLLC
704 - 228th Avenue NE, Suite 193
Sammamish, WA 98074
(425) 836-3030
(425) 836-8957 (facsimile)